

Compresión de imágenes usando Análisis de Componentes Principales

Roi Naveiro y David Ríos Insua

23/06/2021

En este ejercicio se propone aplicar PCA para comprimir imágenes.

1. Descargar datos de <https://drive.google.com/file/d/14f9gJ3SrT2zE8iokixzZBOalvZ4U7YGg/view?usp=sharing>
2. Descomprimir y cargarlos mediante

```
library(EBImage)

# Descargar datos desde
# y descomprimir, luego poner ruta en path:
faces_files <- list.files(path = "data/thumbnails128x128", pattern = ".png",
                          all.files = TRUE, full.names = TRUE, no.. = TRUE)

faces <- readImage(faces_files)
```

3. Representar algunas imágenes

```
plot(faces[,,,11])
```



4. Aplicar PCA

```
faces_flat <- t(array(faces[,,,1:500], dim=c(128^2*3, 500)))
# 2 mins con alrededor de 5000 caras
proy_faces <- prcomp(faces_flat[, 1:(128^2*3)], center = FALSE)
```

5. Encontrar el número de componentes necesarias para que el error de reconstrucción sea como mucho del 1%.

```
eigvals <- proy_faces$sdev^2
ratio <- eigvals / sum(eigvals)
ratio_acum <- cumsum(ratio)
```

```
M = which(ratio_acum >= 0.99)[1]
```

```
print(M)
```

```
## [1] 246
```

```
faces_recons_flat <- proy_faces$x[,1:M] %*% t(proy_faces$rotation[,1:M])
faces_recons_flat[faces_recons_flat<0] <- 0
faces_recons_flat[faces_recons_flat>1] <- 1
```

```
faces_recons <- array(t(faces_recons_flat), dim=dim(faces))
fs = Image(faces_recons, colormode = 'Color')
plot((fs[,,,11]))
```

